IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
Before the Board of Patent Appeals and Interferences

In re Patent Application of

Patrick J. LINK — Atty. Ref.: 723-1443

Serial No. 10/690,818 — TC/A.U.: 3714

Filed: October 23, 2003 — Examiner: David W. DUFFY

For: HAND-HELD VIDEO GAME PLATFORM EMULATION

* * * * * * * * * * *

March 23, 2009 (Monday)

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450
Sir:

REPLY BRIEF

Appellant respectfully submits this Reply to the January 22, 2009 Examiner's Answer. While Appellant appreciates the Examiner's additional comments and analysis, the Examiner has still failed to meet his burden of making a prima facie case of obviousness under 35 U.S.C. § 103(a) for at least the following reasons.

The Examiner concedes that the "Snes9x" primary reference ("The Portable Super Nintendo Entertainment System Emulator v1.19 ReadMe.txt") does not disclose duplicating at least a portion of said allocated ROM pages across said ROM pages as required by independent claims 6, 14 and 37 herein, but asserts that Dahl et al (U.S. Patent No. 5,949,985) overcomes this deficiency and supplies the missing teachings. Dahl et al actually teaches away. See *KSR v. Teleflex*, 550 US___, 127 S.CT. 1727 (2007) ("When the prior art teaches away from combining certain know elements, discovery of successful means of combining them is more likely to be non-obvious.")

The Examiner's Reliance On Dahl et al Is Misplaced

In the Answer, the Examiner indicates that Dahl et al discloses a method and data processing system for emulating a program that utilizes a technical paging architecture that swaps the pages between a main store (e.g., RAM) and a DASD (e.g., hard disk). (see page 5, lines 4-6). The Examiner further asserts that Dahl et al discloses that to make the process more efficient and minimize overhead during emulation, the data is accessed identically, which the Examiner contends minimizes paging. (see page 5, lines 7-8). The Examiner speculates "Since the emulated DASD and emulated main store are accessed identically, it is reasonable to assume that they contain the same files." (see page 5, lines 8-10).

The exact opposite is true. Dahl et al specifically teaches away from storing multiple copies of the same information, and instructs one skilled in the art to store only a single copy. Beginning on line 45 of column 2, Dahl et al states "[a]ccording to the present invention, the emulator accesses instructions of the program directly from the emulated mass storage data area to minimize emulation overhead." (emphasis added). This statement from Dahl et al thus directly contradicts the Examiner's speculation that Dahl et al's emulated main store and emulated DASD store contain the same files.

Dahl et al Teaches AWAY From Duplicating Data In Emulated Memory

Dahl et al's goal is to run programs written for the IBM System/36 (S/36) computer (see Figure 3) on a different computer, namely an IBM AS/400 server 12 illustrated in Fig. 2. Because the AS/400 and System/36 architectures are incompatible,

Dahl et al programs the AS/400 to emulate the System/36 including the System/36 processor (MSP), disk drive (DASD), main store (RAM) and other resources.

In more detail, referring to Fig. 2, the target IBM AS/400 server 12 includes CPUs 30, a main memory 34 and a secondary storage 40. An emulator 38 for emulating the System/36 processor complex 50 is stored in the AS/400's main memory 34. The AS/400 main memory 34 is used to emulate both the System/36 DASD (disk drive) 39 and System/36 main memory store 37.

The System/36 processor cannot access instructions stored in DASD directly; it must first be read into memory before its processor can execute the instructions. To avoid overhead, Dahl et al takes a different approach: "[a]ccording to a second aspect of the present invention, the data processing of the present invention emulates instructions directly out of a simulated DASD backing store in order to minimize emulation overhead." (see column 11, line 67 – column 12, line 4).

Dahl et al takes advantage of the fact that the emulated DASD 39 resides in the memory 34. Thus, instead of emulating the process to page (i.e., swap) the program 46 from the emulated DASD 39 to the emulated main store 37 and then to the emulated MSP 52 for execution, the emulator 38 directly accesses the emulated DASD 39 to retrieve the program 46 instructions. This is possible because in the AS/400 system 12, the CPUs 30 can directly access the memory 34; that is, the CPUs 30 executing the emulator 38 can directly access both the emulated main store 37 and the emulated DASD 39 residing in the memory 34. (see column 5, lines 47-65).

Dahl et al thus specifically teaches that because the program 46 instructions are read directly from the emulated DASD, there is no need to page a copy of portions of

the program 46 in the emulated main store 37. Dahl et al states "according to an important aspect of the present invention, when program 46 is emulated, emulator 38 accesses instructions within program 46 directly out of simulated DASD 39 and does not page the program instructions into simulated main store 37." (emphasis added; see column 4, lines 45-50). The thrust of Dahl et al is to minimize swapping, not by duplicating, but by having zero redundancy of the program 46 to be executed in the emulated system 50.

Simply put, Dahl et al's approach is the exact opposite of Appellant's approach. Whereas Appellant claims duplicating at least a portion of said allocated ROM pages across said ROM pages, Dahl et al's approach is to avoid any such duplication by directly accessing the single copy of information stored in the emulated DASD 39.

In the "Response to Arguments" portion (pages 23-24) of the Answer, the Examiner sets forth a reasoning that one of ordinary skill would allegedly go through to ultimately arrive at Appellant's feature of duplicating at least a portion of an emulated ROM pages across the ROM pages as recited in independent claims 6, 14 and 37. The Examiner's reasoning can perhaps be summarized as follows:

1. Dahl et al discloses a virtualization of a paging system for an emulated system whereby all other data from a fixed and a paged section of storage for an emulated system is stored in acting memory including the emulated page memory.
2. Actual prior art GAMEBOY system ROM memory is composed of fixed and paged sections.[1]

[1] A typical prior art GAMEBOY memory cartridge includes a ROM memory section divided into a plurality of banks. The lower bank (Bank 0) is fixed. The other banks (Bank 1 – Bank n) are bank switched so the GAMEBOY processor can access only one of these upper banks at a time. The cartridge can also contain a memory bank controller to select one of the upper banks, and a RAM.

3. One of ordinary skill would realize that there only two options to emulate this paging system.
   a. Copy the video game ROM image as is and emulate the virtual paging system. This uses less memory but requires additional address translation for the emulating system to be able to access the virtualized paged sections and the emulated fixed portions of code because the virtualized pages would not be in the memory locations that were expected in relations to the fixed portions of the code; OR
   b. Copy the fixed sections of the code to be included with each of the paged sections of memory to preserve the expected layout of the emulated system's memory space. This requires more memory, but reduces the address translation overhead by maintaining the expected arrangement of the memory space and reduces the amount of searching in memory to determine the location of paged and fixed sections.
4. Given the two options and Dahl et al's suggestion to reduce the address translation, it would be obvious to copy the fixed portions of ROM data with paged sections to preserve the expected layout – i.e., it would be obvious to choose option b.

The main problem with the Examiner's analysis is that Dahl et al teaches directly away from his conclusion. In particular, as explained above, Dahl et al teaches NOT duplicating the contents of emulated disk store in emulated main memory, but rather storing only a single copy.

In addition, regarding "1", the Examiner appears to be contending that Dahl et al discloses an emulating system that emulates System/36 paging. This is inaccurate. As noted, Dahl et al specifically states that there is no paging of program from the emulated DASD to the emulated main store. (see column 4, lines 45-50). This directly contradicts the Examiner's contention.

Regarding "3" and "4", the Examiner's allegation that there are only two options is incorrect. Dahl et al actually teaches a third option which it implements to improve performance. Namely, instead of emulating the paging process between the emulated

DASD and the main store, Dahl et al teaches executing program instructions directly from the emulated DASD. This is performed without duplicating portions of the program and still achieves the goal of reducing address translation, which the Examiner puts forth as the motivation for choosing option "b". One of ordinary skill, when provided with explicit teachings in Dahl et al, would choose neither option "a" nor "b" but would instead follow Dahl et al's teachings.

The Examiner's statements regarding options "a" and "b" are also problematic. Regarding option "a", the Examiner suggest that copying the ROM data "as is" would lead to virtualized pages being in locations that are unexpected in relation to the fixed portions of the code. This is a self-contradictory statement. It is difficult to see how faithfully copied ROM data would be in unexpected locations relative to each other. Examiner's statement regarding option "b" is also self-contradicting in that it suggests deviating from the original order in copying the ROM data somehow leads to "expected" locations.

For at least these reasons, Dahl et al cannot be relied upon to teach or suggest duplicating portions of the emulated ROM pages. The Examiner's *prima facie* case of obviousness fails.

Dahl et al Also Does Not Teach Appellant's Claimed Pointer Table System

Dependent claim 23 requires in combination, using said pointer table system to control memory access by remapping memory access instructions into function call. Claim 6 also requires a pointer table system. On page 7 of the Answer, the Examiner

asserts that column 5, lines 11-43 of Dahl et al disclose utilizing translation or paged tables. Dahl et al does indeed teach using an address translation table:

> "The address translation table associates the page selector bits with a 16-bit page number, which is concatenated to the page offset (bits 13-23 of the address) to access the appropriate physical memory location. By referencing storage objects through the address translation table, the memory pages associated with a storage object can be scattered in discontiguous physical locations within main store 58 while appearing to be contiguous to application programs. It is important to note that address translation is performed during each memory access which utilizes translated addressing mode." (see column 5, lines 32-40).

This section of Dahl et al is outside the context of emulation and merely describes the address translation that exists in typical paging architectures (see lines 24-25) so that storage objects need not be stored contiguously in real memory. There is no teaching of using pointer tables to remap memory access instructions into function calls as Appellant has claimed. The Examiner does not articulate any reasoning to support his conclusion of obviousness.

Within the emulation context, Dahl et al states "More importantly, since all program objects are stored contiguously within a single segment (i.e., the segment allocated to simulated DASD 39), address translation overhead is eliminated when fetching emulated instructions within program 46." (emphasis added; see column 5, lines 54-58). This directly teaches against remapping memory access instructions into function calls.

Appellant respectfully maintains that all remaining claims are not obvious in view of the cited references based on their dependencies from independent claims as well as on their own, as articulated in the Appeal Brief.

For at least the reasons stated above, the Examiner's rejection must be reversed and all claims passed to issue.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: _____
Hyung N. Sohn
Reg. No. 44,346

HNS/edg
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100